

IN THE ABSTRACT:

Please substitute the following annotated paragraph for the abstract:

Described are systems and methods for scheduling threads for a multi-tasking operating system. In accordance with some embodiments, a rate-based scheduling algorithm is incorporated to provide a flexible set of Quality of Service guarantees controlling the allocation of CPU to a thread, together with a mechanism to provide latency control for hard real time software. Additionally, the present invention is particularly suitable for use in systems where essentially all software runs under the control of the scheduler, from "interrupt handler" functions, through multimedia applications to simple "console" functions and non-real time tasks. In an additional embodiment, the system and method of the present invention also actively limits the CPU time allocated to a given software thread to defend against malicious or accidental denial of service attacks. A circular array structure is maintained having multiple time slots, where each time slot corresponds to a timeslice during which CPU resources are allocated to a particular thread. The time slots in the circular array include a queue of threads scheduled for execution during that time slot. A pointer index and an array of threads requesting immediate CPU resource allocation are maintained. A currently executing thread is suspended, and a next time slot during which the currently executing thread should resume execution is calculated. The suspended currently executing thread is appended to the queue of threads scheduled for execution at the calculated time slot. The pointer index is updated to point to the identified next sequential non-empty time slot. Any contents of the indexed time slot is appended to the array of threads requesting immediate CPU resource allocation. The thread at the top of the array is removed and activated.